Subject: Array problems
Posted by rafiwui on Wed, 09 Aug 2017 11:01:27 GMT
View Forum Message <> Reply to Message

I have 2 questions concerning arrays:
1. Why can't I add my member to my array like this:

```
Array<ParentCtrl> array;
WithTestLayout<ParentCtrl> testLayout;
array.Add(testLayout);
```

If I try this the compiler is not amused so I have to do it like this:

```
array.Add(&testLayout);
```

But when I tried to reproduce it with an Array<int> I can use the first snippet.

2. I did it with the working approach but when I close the app it says this:

```
Exception: C0000005 at     7FF727395F8A
EXCEPTION_ACCESS_VIOLATION
reading at 36705825540
```

This exception isn't thrown when I reproduced the error in the attached project. Instead there is an Heap is corrupted error.

I encountered both of these things as well when I made an array of a class I created myself.

Any idea or hints for how I get this to work properly?

File Attachments

1) Bugtester.7z, downloaded 314 times

Subject: Re: Array problems
Posted by Oblivion on Wed, 09 Aug 2017 11:34:35 GMT
View Forum Message <> Reply to Message

Hello,

Answer to question 2 (your first question is closely related to your second question): You are deallocating an already deallocated memory (testLayout). Because you've passed the address of a stack allocated object to Array (which is nor forbidden since Array does not know, nor does it care how, or from where (stack or heap), its elements are created as long as you give its address. It allows taking the ownership (of HEAP allocated objects). Read the docs Luke! :)). But you forget that when the array gets destroyed, so is testlayout. So you are also corrupting the program's heap, since testLAyout is allocated from stack, and "&layout" gets "delete[ed]" which is a valid operation only on heap-allocated objects.

Did you try:

```
Array<WithTestLayout<ParentCtrl> > layouts;
auto& testLayout = layouts.Add(); // <-- Creates the ParentCtrl with the given layout as its element.
```

Best regards,
Oblivion

Subject: Re: Array problems
Posted by rafiwui on Wed, 09 Aug 2017 11:55:42 GMT

View Forum Message <> Reply to Message

Quote:Answer to question 2 (your first question is closely related to your second question):
I thought so.

Quote:
Did you try:

```
Array<WithTestLayout<ParentCtrl> > layouts;
auto& testLayout = layouts.Add(); // <-- Creates the ParentCtrl with the given layout as its element.
```

This would be a solution for an array full of a specific layout, but I want an array where I can put different layouts and call them depending on some other things. And with this approach all the elements in the array are of type Array<WithTestLayout<ParentCtrl>>.

Subject: Re: Array problems
Posted by Oblivion on Wed, 09 Aug 2017 13:18:20 GMT

View Forum Message <> Reply to Message

Hello,

That was a missing information. Next time try to be a little more specific. :)

Anyway, here you go:

```
TestWindow::TestWindow()
{
	// Array<ParentCtrl> layoutArray;
```

```
        // Note that we're using static_cast here, for the sake of simplicity. When you have several layouts in same array, using One<>, which has RTTI support, or dynamic_cast would be better.

 auto& t = layoutArray.Add(new WithTestLayout<ParentCtrl>); // <-- Array "owns" the new ParentCtrl with layout. No need to delete explicitly.
 CtrlLayout(static_cast<WithTestLayout<ParentCtrl>&>(t));
        Add(t.SizePos());

}
```

I hope this will help.

Best regards,
Oblivion

Subject: Re: Array problems
Posted by rafiwui on Wed, 09 Aug 2017 13:27:05 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Wed, 09 August 2017 15:18Hello,
That was a missing information. Next time try to be a little mode specific. :)

Sorry for that.

Working fine now, but is something wrong with SizePos?

Should look like this:

Somehow it only resizes vertical :lol:

File Attachments
1) bugtester.PNG, downloaded 663 times
2) bugtester2.PNG, downloaded 622 times

Subject: Re: Array problems
Posted by rafiwui on Wed, 09 Aug 2017 13:29:07 GMT
View Forum Message <> Reply to Message

Ah never mind I realised that I forgot to set the horizontal size spring x(

Subject: Re: Array problems
Posted by rafiwui on Thu, 10 Aug 2017 14:21:18 GMT

View Forum Message <> Reply to Message

By the way: I found an even better/safer method of doing this (IMO):
Create a class inheriting from the layout so you don't have to create the auto& variables:

```
class TestLayout : public WithTestLayout<ParentCtrl>
{
public:
    typedef TestLayout CLASSNAME;
    TestLayout() { CtrlLayout(*this); }
};

TestWindow::TestWindow()
{
    Array<ParentCtrl> layoutArray;
    layoutArray.Add(new TestLayout);
    Add(layoutArray[0].SizePos());
}
```

Subject: Re: Array problems
Posted by mirek on Sat, 26 Aug 2017 11:11:58 GMT

View Forum Message <> Reply to Message

Using 'new' is fine, but even safer is to use "Create":

```
layoutArray.Create<TestLayout>();
```

Generally, I believe in not exposing new, delete, and heap pointers.. (of course within reason).

Mirek